

IBM Docket No. AUS920011024US1

1

## TITLE OF THE INVENTION

Method and System for Performance Reporting in a Network Environment

## 5 CROSS-REFERENCES TO RELATED APPLICATIONS, AND COPYRIGHT NOTICE

10 The present application is related to a co-pending application entitled Method and System for Probing in a Network Environment, filed on even date herewith, assigned to the assignee of the present application, and herein incorporated by reference. A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

## 15 FIELD OF THE INVENTION

20 The present invention relates generally to information handling, and more particularly to methods and systems for evaluation of the performance of information handling in a network environment, and for reporting performance data.

## BACKGROUND OF THE INVENTION

25 Various approaches have been proposed for monitoring, simulating, or testing web sites. Examples include U.S. Pat. No. 6,278,966 B1 (Howard, et al., Aug. 21, 2001), "Method and System for Emulating Web Site Traffic to Identify Web Site Usage Patterns." However, this example addresses substantially different problems (problems of simulation and hypothetical phenomena), and thus is significantly different from the present invention. Other examples include U.S. Pat. No. 6,078,956 (Bryant, et al., Jun. 30, 2000) and U.S. Pat. No. 5,787,254 (Maddalozzo, et al.,

5 Jul.28, 1998). Other examples include services available from vendors such as Atesto Technologies Inc., Keynote Systems, and Mercury Interactive Corporation. These services may involve a script that runs on a probe computer. The examples mentioned above do not necessarily provide for some comparisons that are useful for problem-solving.

10 A wide variety of valuable services are provided through client-server applications, so proper performance of client-server applications may be very important. Lack of useful performance information can hamper problem-solving efforts for client-server applications. Thus there is a need for systems and methods that effectively communicate performance information regarding client-server applications, including but not limited to web sites.

#### 15 SUMMARY OF THE INVENTION

20 An example of a solution to problems mentioned above comprises: collecting data from a plurality of probes, including at least one local probe and at least one remote probe; and reporting the data. For example, the reporting may comprise: reporting a first subset of the data that originates from a local probe; reporting a second subset of the data that originates from remote probes; and employing a similar reporting format for said first subset and said second subset. Thus comparison of data from a local probe and data from remote probes may be facilitated.

25 To give further examples of the solutions provided, there are solutions comprising: receiving data from at least one probe; comparing said data with at least one threshold value derived from a service level agreement; and reporting results of said comparing. Another example comprises: receiving data from at least one probe; comparing said data with at least one threshold

value derived from a service level agreement; and outputting in a special mode any measured response time value that is greater than the corresponding threshold value.

5 BRIEF DESCRIPTION OF THE DRAWINGS

A better understanding of the present invention can be obtained when the following detailed description is considered in conjunction with the following drawings. The use of the same reference symbols in different drawings indicates similar or identical items.

10 FIG. 1 illustrates a simplified example of a computer system capable of performing the present invention.

FIG. 2 is a block diagram illustrating one example of how the present invention was implemented for web site measurement.

15 FIG. 3A and FIG. 3B illustrate an example of a report that was generated with data from remote probes.

FIG. 4A and FIG. 4B illustrate an example of a report that was generated with data from a local probe.

20 FIG. 5A and FIG. 5B illustrate an example of a report that was generated with data from a local probe, with error messages.

DETAILED DESCRIPTION

25 The examples that follow involve the use of one or more computers and may involve the use of one or more communications networks.

The present invention is not limited as to the type of computer on which it runs, and not limited as to the type of network used.

30 The following are definitions of terms used in the description of the present invention and in the claims:

"Availability" means ability to be accessed or used.

"Client-server application" means any application involving a client that utilizes a service, and a server that provides a service. Examples of such a service include but are not limited to: information services, transactional services, access to databases, and access to audio or video content.

"Comparing" means bringing together for the purpose of finding any likeness or difference, including a quantitative likeness or difference. "Comparing" may involve answering questions including but not limited to: "Is a measured response time greater than a threshold response time?" Or "Is a response time measured by a remote probe significantly greater than a response time measured by a local probe?"

"Computer-usable medium" means any carrier wave, signal or transmission facility for communication with computers, and any kind of computer memory, such as floppy disks, hard disks, Random Access Memory (RAM), Read Only Memory (ROM), CD-ROM, flash ROM, non-volatile ROM, and non-volatile memory.

"Measuring" means evaluating or quantifying.

"Outputting" means producing, transmitting, or turning out in some manner, including but not limited to printing on paper, or displaying on a screen, or using an audio device.

"Performance" means execution or doing; "performance" may refer to any aspect of an application's operation, including availability, response time, time to complete batch processing or other aspects.

"Probe" means any computer used in evaluating, investigating, or quantifying performance; for example a "probe" may be a personal computer executing a script, acting as a client, and requesting services from a server.

"Response time" means elapsed time in responding to a request or signal.

"Script" means any program used in evaluating, investigating, or quantifying performance; for example a script may cause a computer to send requests or signals according to a transaction scenario. A script may be written in a scripting language such as Perl or some other programming language.

"Service level agreement" means any oral or written agreement between provider and user. For example, "service level agreement" includes but is not limited to an agreement between vendor and customer, and an agreement between an information technology department and an end user. For example, a "service level agreement" might involve one or more client - server applications, and might include specifications regarding availability, response times or problem - solving.

"Storing" data or information, using a computer, means placing the data or information, for any length of time, in any kind of computer memory, such as floppy disks, hard disks, Random Access Memory (RAM), Read Only Memory (ROM), CD-ROM, flash ROM, non-volatile ROM, and non-volatile memory.

"Threshold value" means any value used as a borderline, standard, or target; for example, a "threshold value" may be derived from a service level agreement, industry norms, or other sources.

FIG. 1 illustrates a simplified example of an information handling system that may be used to practice the present invention. The invention may be implemented on a variety of hardware platforms, including personal computers, workstations, servers, and embedded systems. The computer system of FIG. 1 has at least one processor 110. Processor 110 is interconnected via system bus 112 to random access memory (RAM) 116, read only memory (ROM) 114, and input/output (I/O) adapter 118 for connecting peripheral devices such as disk unit 120 and tape drive 140 to bus 112. The system has user interface adapter 122

for connecting keyboard 124, mouse 126, or other user interface devices such as audio output device 166 and audio input device 168 to bus 112. The system has communication adapter 134 for connecting the information handling system to a data processing network 150, and display adapter 136 for connecting bus 112 to display device 138. Communication adapter 134 may link the system depicted in FIG. 1 with hundreds or even thousands of similar systems, or other devices, such as remote printers, remote servers, or remote storage units. The system depicted in FIG. 1 may be linked to both local area networks (sometimes referred to as Intranets) and wide area networks, such as the Internet.

While the computer system described in FIG. 1 is capable of executing the processes described herein, this computer system is simply one example of a computer system. Those skilled in the art will appreciate that many other computer system designs are capable of performing the processes described herein.

FIG. 2 is a block diagram illustrating one example of how the present invention was implemented for web site measurement. As an overview, this example comprised: collecting data (in databases shown at 222 and 251) from a plurality of probes, including at least one local probe (shown at 221) and at least one remote probe (shown at 235); and reporting said data (reports shown at 241 and 242). The reporting further comprised: reporting a first subset (report shown at 241) of the data that originated from a local probe 221; reporting a second subset (report shown at 242) of the data that originated from remote probes 235; and employing a similar reporting format for said first subset and said second subset. Thus comparison of data from a local probe 221 and data from remote probes 235 was facilitated. The double-headed arrow

at 250 symbolizes comparison. Such comparison proved to be useful for solving performance problems.

5 The same script was deployed on the local and remote probes. The local probe 221 provided information that excluded the Internet, while the remote probes 235 provided information that included the Internet (shown at 290). Thus, the information could be compared to determine whether performance or availability problems were a function of the site itself (infrastructure - specific or application - specific), or a function of the Internet 290.

10 Another aspect of this example was comparing data (e.g. data stored in database 222) with at least one threshold value derived from a service level agreement (symbolized by "SLA specs" at 15 262); and reporting results (e.g. in report 242) of said comparing.

20 Turning now to some details of the example implementation, probes measured response time for requests. The double-headed arrow connecting remote probes at 235 with application 201 symbolizes requests and responses. The double-headed arrow connecting local probe 221 with application 201 symbolizes requests and responses.

25 We located application probes locally at hosting sites (e.g. local probe shown at 221, within data center 211) and remotely at relevant end-user sites (remote probes at 235). This not only exercised the application code and application hosting site 30 infrastructure, but also probed the ability of the application and network to deliver data from the application hosting site to the remote end-user sites. End-to-end measurement of IBM external

applications (symbolized by application 201 with web pages 202) for customers or business partners, for example, involved remote application probes (RAP's) on the Internet (remote probes at 235 shown within Internet 290). While we measured the user availability and performance from a customer perspective (remote probes at 235), we also measured the availability and performance of the application at the location where it was deployed (local probe shown at 221, within data center 211). This provided baseline performance measurement data, that could be used for analyzing the performance measurements from the remote probes (at 235).

Local probe 221 was implemented with a personal computer, utilizing IBM's Enterprise Probe Platform technology, but other kinds of hardware and software could be used. A local probe 221 was placed on the IBM network just outside the firewall at the center where the web site was hosted. A local probe 221 was used to probe one specific site per probe. There could be multiple scripts per site. A local probe 221 executed the script every 20 minutes. Intervals of other lengths also could be used.

Another aspect of this example was providing an alert when data indicated an error. An example of an error is a measured response time value greater than a corresponding threshold value. For example, if a local probe 221 encountered a problem (e.g. it was unable to access the site or unable to complete the script) on two consecutive executions of the script, local probe 221 generated a real time alert (problem event), and sent it to a TIVOLI management system (shown as console 205). Another similar kind of management system could be used. Thus an alert was provided via a system management computer. An alert message via email also could be used.



Local probe 221 sent to a database 251 the data produced by the measuring process. Database 251 was implemented by using IBM's DB2 technology, but other database management software could be used, such as ORACLE, INFORMIX, SYBASE, MYSQL, Microsoft Corporation's SQL SERVER, or similar software. For local probe data, an automated reporting tool (shown as report generator 231) ran continuously at set intervals, obtained data from database 251, and sent reports 241 via email to these IBM entities: the web site owner, the hosting center, and IBM's world wide command center. Reports 241 also could be posted on a web site at the set intervals. Report generator 231 was implemented by using the Perl scripting language and the AIX operating system. However, some other programming language could be used, and another operating system could be used, such as LINUX, or another form of UNIX, or some version of Microsoft Corporation's WINDOWS, or some other operating system. Note that in an alternative example, report generator 231 might obtain data from databases at 251 and at 222, then generate reports 241 and 242.

Remote probes at 235 were implemented by contracting for probing services available from Mercury Interactive Corporation, but services from another vendor could be used, or remote probes could be implemented by other means (e.g. directly placing probes at various ISP's). A remote probe 235 may be used to probe one specific site per probe; a probe also has the capability of probing multiple sites. There could be multiple scripts per site. Remote probes 235 were located at various ISP's in parts of the world that the web site supported. A remote probe 235 executed the script every 60 minutes. Intervals of other lengths also could be used. If multiple remote probes at 235 are used, probe execution times may be staggered over the hour to ensure that the

performance of the web site is being measured throughout the hour. Remote probes at 235 sent to a database 222 the data produced by the measuring process. Database 222 was implemented by using Mercury Interactive's database, but other database management software could be used, such as IBM's DB2, ORACLE, INFORMIX, SYBASE, MYSQL, Microsoft Corporation's SQL SERVER, or similar software. Report generator 232 was implemented by using Mercury Interactive's software and web site, but another automated reporting tool could be used, such as the one described above for local probe data (shown as report generator 231). IBM's arrangement with Mercury Interactive included the following: Mercury Interactive's software at 232 used IBM's specifications (symbolized by "SLA specs" at 262) and created near-real-time reports (symbolized by report 242) in a format required by IBM; IBM's specifications and format were protected by a confidential disclosure agreement; the reports at 242 were supplied in a secure manner via Mercury Interactive's web site at 232; access to the reports was restricted to IBM entities (the web site owner, the hosting center, and IBM's world wide command center).

Turning now to some details of collecting data from a plurality of probes, Component Probes measure availability, utilization and performance of infrastructure components, including servers, LAN, and services. Local component probes (LCPs) may be deployed locally in hosting sites, service delivery centers or data centers (e.g. at 211).

Network Probes measure network infrastructure response time and availability. Remote Network Probes (RNPs) may be deployed in a local hosting site or data center (e.g. at 211) if measuring the intranet or at Internet Service Provider (ISP) sites if measuring the Internet.

Application Probes measure availability and performance of applications and business processes.

Local Application Probe (LAP): Application probes deployed in a local hosting site or data center (e.g. at 211) are termed Local Application Probes.

Remote Application Probe (RAP): An application probe deployed from a remote location is termed a Remote Application Probe.

The concept of "probe" is a logical one. Thus, implementing a local component probe could actually consist of implementing multiple physical probes.

Providing a script for a probe would comprise defining a set of transactions that are frequently performed by end users.

Employing a plurality of probes would comprise placing at least one remote probe (shown at 235 in FIG. 2) at each location having a relatively large population of end users. Note that the Remote Application Probe transactions and Local Application Probe transactions should be the same transactions. The example measured all the transactions locally (shown at 221), so that the local application response time can be compared to the remote application response time. (The double-headed arrow at 250 symbolizes comparison.) This can provide insight regarding application performance issues. End-to-end measurement of an organization's internal applications for internal customers will involve a RAP on an intranet, whereas end-to-end measurement of an organization's external applications for customers, business partners, suppliers, etc. will involve a RAP on the Internet (shown at 235). The example involved defining a representative transaction set, and deploying remote application probes at relevant end-user locations. (This simplicity is something that can only be appreciated when this example is contrasted with

other more complicated models.) A benefit following from the simplicity of this example is that it is easily generalized to other environments besides web based applications. Application 201 may be any client-server application. Some examples are a web site, a web application, database management software, a customer relationship management system, an enterprise resource planning system, or an opportunity - management business process where a client directly connects to a server.

FIG. 3A and FIG. 3B illustrate an example of a report that was generated with data from remote probes. The data was produced by probing a web site that served an after-sales support function. The probes used a script representing a typical inquiry about a product warranty. Similar reports could be produced in connection with probing other kinds of web sites, or probing other kinds of client-server applications.

To begin with an overview, the broken line AA shows where the report is divided into two sheets. Columns 303 - 312 display response time data in seconds. Each of the columns 303 - 311 represent a transaction step in a script. Column 312 represents the total of the response times for all the transaction steps. A description of the transaction step is shown in the column heading in row 321. Column 313 displays availability information, using a color code. In one example, the cell in column 313 was green if all the transaction steps were completed; otherwise the cell was red, representing a failed attempt to execute all the transaction steps. Thus column 313 may provide a measure of end-to-end availability from a probe location, since a business process could cross multiple applications deployed in multiple hosting centers. Column 302 shows probe location and Internet service provider information. Column 301 shows time of script

execution. Each row from row 323 downward represents one iteration of the script; each of these rows represents how one end user's execution of a business process would be handled by the web site. Cells with a special color display service level agreement violations; in this example, a special color is shown by darker shading, seen in the cells of column 311 for example. A report may contain error messages (not shown in this example).

Turning to some details of FIG. 3A and FIG. 3B, this example involves comparing data with threshold values derived from a service level agreement. To report the results of this comparing, color is used in this example. Row 322 shows threshold values. In each column, response times for a transaction step are compared with a corresponding threshold value. For example, column 303 is for the "open URL" step. For that step, column 303 reports results of each script execution by a plurality of probes. Reporting involves outputting in a special mode any measured response time value that is greater than the corresponding threshold value. Outputting in a special mode may mean outputting in a special color, for example, or outputting with some other visual cue such as highlighting or a special symbol. A special color is shown by darker shading, seen in the cell at 333 for example. In one implementation, the special color was red, and other cells were in green. In this example, hypertext markup language (HTML) was used to create the report, but another language such as extensible markup language (XML) could be used.

FIG. 4A and FIG. 4B illustrate an example of a report that was generated with data from a local probe. This example may be considered by itself as an example involving one probe, or may be considered together with the example shown in FIG. 3A and FIG. 3B. The data was produced by probing the web site that was

described above regarding FIG. 3A and FIG. 3B. The script also is described above. Again, similar reports could be produced in connection with probing other kinds of web sites, or probing other kinds of client-server applications.

5

To begin with an overview, the broken line AA shows where the report is divided into two sheets. This example comprised: receiving data from a probe; comparing said data with threshold values (row 422) derived from a service level agreement; and reporting results of said comparing. Columns 403 - 412 display response time data in seconds. Each of the columns 403 - 411 represent a transaction step in a script. Column 412 represents the total of the response times for all the transaction steps. A description of the transaction step is shown in the column heading in row 421. Column 413 displays availability information, using a color code. In one example, the cell in column 413 was green if all the transaction steps were completed; otherwise the cell was red, representing a failed attempt to execute all the transaction steps. Thus column 413 may provide a measure of end-to-end availability from a probe location, since a business process could cross multiple applications deployed in multiple hosting centers. Column 402 shows probe location. Column 401 shows time of script execution. Each row from row 423 downward represents one iteration of the script; each of these rows represents how one end user's execution of a business process would be handled by the web site. Cells with a special color display service level agreement violations (not shown in this example). A report may contain error messages (not shown in this example).

10

15

20

25

30

Turning to some details of FIG. 4A and FIG. 4B, this example involves comparing data with threshold values derived from a

service level agreement. To report the results of this comparing, color is used in this example. Row 422 shows threshold values. In each column, response times for a transaction step are compared with a corresponding threshold value. For example, column 403 is for the "open URL" step. For that step, column 403 reports results of each script execution.

The reporting comprised: reporting a subset (report shown in FIG. 4A and FIG. 4B) of the data that originated from a local probe; reporting a subset (report shown in FIG. 3A and FIG. 3B) of the data that originated from remote probes; and employing a similar reporting format for both subsets. Thus comparison of data from a local probe and data from remote probes was facilitated.

Regarding threshold values, note that an alternative example might involve threshold values that differed between the local and remote reports. Threshold values may need to be adjusted to account for Internet - related delays. Also note that FIG. 4A and FIG. 4B illustrate an example where hypertext markup language (HTML) was used to create the report, but another language such as extensible markup language (XML) could be used.

FIG. 5A and FIG. 5B illustrate an example of a report that was generated with data from a local probe, with error messages. This example may be considered by itself as an example involving one probe, or may be considered together with the examples shown in FIG. 3A, FIG. 3B, FIG. 4A, and FIG. 4B. The data was produced by probing a web site. The script was similar to the script described above. Again, similar reports could be produced in connection with probing other web sites, or probing other kinds of client-server applications. Again, the broken line AA shows where the report is divided into two sheets. The column on the far left shows time of script execution. The column on the far

right displays availability information, using a color code. In this example, reporting comprised outputting in a special mode an indication of an application's lack of availability. Outputting in a special mode may mean outputting in a special color, for example, or outputting with some other visual cue such as highlighting or a special symbol. A special color is shown by darker shading, seen in the middle rows (with execution times from 02:58:44 to 04:23:10) for example. In one implementation, the special color was red, and other cells were in green.

FIG.'s 5A and 5B also provide an example of receiving data from at least one probe; comparing said data with at least one threshold value derived from a service level agreement; and outputting in a special mode any measured response time value that is greater than the corresponding threshold value. Here, a special mode was a special color, shown by the darker shading seen in the middle rows (with execution times from 02:58:44 to 04:23:10) for example. The affected rows also contain an error message to guide problem - solving efforts: "Application Error: Logon Failed." In this example, hypertext markup language (HTML) was used to create the report, but another language such as extensible markup language (XML) could be used.

Note that the examples shown in FIG.'s 3A, 3B, 4A, 4B, 5A, and 5B involved reporting results of each script execution by a probe. These are examples of comprehensive reporting, rather than displaying summaries or averages. Reporting results of each script execution allows immediate recognition of problems, and provides guidance for immediate problem - solving efforts.

To continue with this example in FIG.'s 5A and 5B, involving error messages, reporting may comprise providing an alert when



the data indicates an error. The alert may be provided via email, for example, or may be provided in real time via a network and a system management computer. A clearing message may be provided when the error no longer is detected. In the example shown in FIG. 2, local probe 221 generated a real time alert (problem event), and sent it to a TIVOLI management system (shown as console 205). Another similar kind of management system could be used.

The local probe automatically sent events to the management console used by the operations department. In the example solution, integration was provided with the TIVOLI MANAGEMENT ENVIRONMENT and the TIVOLI EVENT CONSOLE product. The example solution generated events from the local probe, and the events were generated after two consecutive errors on the same step in the business process. This could then be adjusted to send an event on the first error, for even faster notification. The recommendation is to send events on the second occurrence initially and then adjust to sending the event on the first occurrence as the environment becomes more stable and better understood by the operational staff. The reason for the recommendation is that in a Web environment there are a number of things that can cause intermittent problems, and it is ultimately a business decision when to invoke problem determination procedures. The report example in FIG.'s 5A and 5B shows an example of a condition where an event was generated. There were five consecutive executions failing on step two. The event was generated starting at 02:58:44 and contained information about the failed business process.

The associated event sent to the TIVOLI ENTERPRISE CONSOLE had a severity of "Warning," and documented the failure of step two

IBM Docket No. AUS920011024US1

18

where the probe was unable to log on to the web site. Note that the date and time is the local time from the probe and that the reports could be generated using different time zones, such as GMT or as in this case British Summer Time. In the examples below, "CRT" refers to a type of probe technology used by IBM. An example of an alert follows.

Tivoli alert for CRT probe failure:

Tivoli CRT Alert - PC NA

WARNING (NAQS2 [LogonFailed / 1])

PCNa - Partner Commerce North America CRT Monitor

tag:auth=crtGwaFw

tag:message=PartnerCommerceNA https://ecna.partner.com Step-NAQS2-failed: Logon failed.

tag:severity=WARNING

tag:slot hostname=d03bpec18.pinfo.com

tag:slot mail\_svr=CVRM

tag:slot mta=ecna.partner.com

tag:slot probe\_addr=NAQS2

tag:slot probe\_date=07/21

tag:slot probe\_time=19:58

tag:class=crt\_event

tag:source=SENTRY

The error messages displayed in the report should be customized, interpret browser level messages, and be meaningful to the operational staff, to shorten problem determination time.

It is useful to automatically close opened events if a subsequent business process is executed successfully. This allows the operational staff to direct time and efforts to those events that remain in "open" status. Below is an example of such an event which was used to automatically close the previously opened event. The event was reported as severity HARMLESS and with the appropriate rules defined on the TIVOLI ENTERPRISE CONSOLE the previously opened event would be closed. Referring back to the report example in FIG.'s 5A and 5B, this HARMLESS event was generated when the probe successfully executed the script starting at 4:43:09 and was able to log on to the site. An example of such an event follows.

Tivoli alert for CRT probe failure:

Tivoli CRT Alert - PC NA

HARMLESS (NAQS2 [RecoveredZf / 0])

PCNa - Partner Commerce North America CRT Monitor

tag:auth=crtGwaFw

tag:message=PartnerCommerceNA https://ecna.partner.com Step-NAQS2-failed: The problem causing the previous alert has been fixed.

tag:severity=HARMLESS

tag:slot hostname=d03bpec18.pinfo.com

tag:slot mail\_svr=CVRM

tag:slot mta=ecna.partner.com

tag:slot probe\_addr=NAQS2

tag:slot probe\_date=07/21

tag:slot probe\_time=21:43

tag:class=crt\_event

tag:source=SENTRY

In conclusion, we have shown examples of methods and systems for performance reporting in a network environment.

One of the preferred implementations of the invention is an application, namely a set of instructions (program code) in a code module which may, for example, be resident in the random access memory of a computer. Until required by the computer, the set of instructions may be stored in another computer memory, for example, in a hard disk drive, or in a removable memory such as an optical disk (for eventual use in a CD ROM) or floppy disk (for eventual use in a floppy disk drive), or downloaded via the Internet or other computer network. Thus, the present invention may be implemented as a computer-usable medium having computer-executable instructions for use in a computer. In addition, although the various methods described are conveniently implemented in a general-purpose computer selectively activated or reconfigured by software, one of ordinary skill in the art would also recognize that such methods may be carried out in hardware, in firmware, or in more specialized apparatus constructed to perform the required method steps.

While the invention has been shown and described with reference to particular embodiments thereof, it will be understood by those skilled in the art that the foregoing and other changes in form and detail may be made therein without departing from the spirit and scope of the invention. The appended claims are to encompass within their scope all such changes and modifications as are within the true spirit and scope of this invention. Furthermore, it is to be understood that the invention is solely defined by

